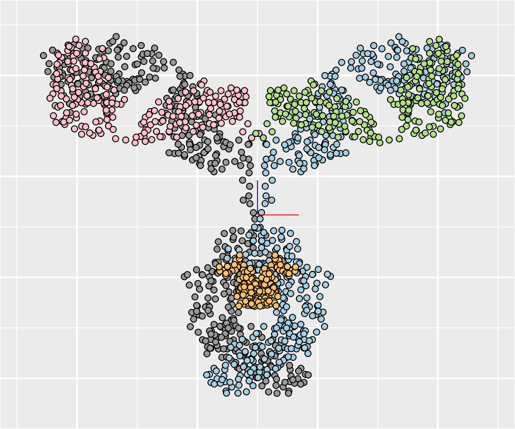
Populations

R.W. Oldford

UNIVERSITY OF WATERLOO

Problem: Structure of human immunoglobulin G1 (IgG1)

Recall exploring how the geometry of the human immunoglobulin G1 molecule related to different variables associated with each “alpha” carbon.



E.g. here, colours are assigned to each carbon atom according to the value of its `chainID` variable.

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

We imagine each alpha carbon as a single **unit** in this set and, because these are **all** the alpha carbons of this molecule, statistically we imagine that the set as the **population** of all the alpha carbons in IgG1.

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

We imagine each alpha carbon as a single **unit** in this set and, because these are **all** the alpha carbons of this molecule, statistically we imagine that the set as the **population** of all the alpha carbons in IgG1.

More abstractly, denote a **unit** by u and the **population** of all units (i.e. alpha carbons) as $\mathcal{P}$.

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

We imagine each alpha carbon as a single **unit** in this set and, because these are **all** the alpha carbons of this molecule, statistically we imagine that the set as the **population** of all the alpha carbons in IgG1.

More abstractly, denote a **unit** by u and the **population** of all units (i.e. alpha carbons) as $\mathcal{P}$. The 1,556 alpha carbons could then be denoted individually as $u_1, u_2, \ldots, u_{1556}$ with $\mathcal{P} = \{u_1, \ldots, u_{1556}\}$.

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

We imagine each alpha carbon as a single **unit** in this set and, because these are **all** the alpha carbons of this molecule, statistically we imagine that the set as the **population** of all the alpha carbons in IgG1.

More abstractly, denote a **unit** by u and the **population** of all units (i.e. alpha carbons) as $\mathcal{P}$. The 1,556 alpha carbons could then be denoted individually as $u_1, u_2, \ldots, u_{1556}$ with $\mathcal{P} = \{u_1, \ldots, u_{1556}\}$.

The data frame `igg1` (from the package loon.data) has 1556 rows, one for each alpha carbon. In the above notation, we can take

- row i to be the ith unit in $\mathcal{P}$, and
- the i'th value of `rownames(igg1)` as u_i.

Note:

- like the u_is, `rownames(igg1)` must be unique;

Problem: Populations and units

There are 1,556 individual carbon atoms which constitute **the entire set** of alpha carbons in the human IgG1 molecule.

We imagine each alpha carbon as a single **unit** in this set and, because these are **all** the alpha carbons of this molecule, statistically we imagine that the set as the **population** of all the alpha carbons in IgG1.

More abstractly, denote a **unit** by u and the **population** of all units (i.e. alpha carbons) as $\mathcal{P}$. The 1,556 alpha carbons could then be denoted individually as $u_1, u_2, \ldots, u_{1556}$ with $\mathcal{P} = \{u_1, \ldots, u_{1556}\}$.

The data frame `igg1` (from the package loon.data) has 1556 rows, one for each alpha carbon. In the above notation, we can take

- row i to be the ith unit in $\mathcal{P}$, and
- the i'th value of `rownames(igg1)` as u_i.

Note:

- like the u_is, `rownames(igg1)` must be unique;
- they can also be thought of as possible **keys** to identify identical units (e.g. as `linkingKeys` in `loon` plots).

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another
- u can be thought of simply as a unique key and so can be represented by any identifiable and unique label for each unit in the population
  - e.g. the row number/label printed out by `head(igg1)`

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another
- u can be thought of simply as a unique key and so can be represented by any identifiable and unique label for each unit in the population
  - e.g. the row number/label printed out by `head(igg1)`
- of course in some cases, it may be easier to identify the population and units **before** identifying individual units with unique labels
  - e.g. $\mathcal{P}$ is the set of all alpha carbon atoms in the molecule IgG1, each alpha carbon being a unit

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another
- u can be thought of simply as a unique key and so can be represented by any identifiable and unique label for each unit in the population
  - e.g. the row number/label printed out by `head(igg1)`
- of course in some cases, it may be easier to identify the population and units **before** identifying individual units with unique labels
  - e.g. $\mathcal{P}$ is the set of all alpha carbon atoms in the molecule IgG1, each alpha carbon being a unit
- for simplicity, we often take $\mathcal{P} = \{1, \ldots, N\}$ where N is the (finite) cardinality of $\mathcal{P}$

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another
- u can be thought of simply as a unique key and so can be represented by any identifiable and unique label for each unit in the population
    - e.g. the row number/label printed out by `head(igg1)`
- of course in some cases, it may be easier to identify the population and units **before** identifying individual units with unique labels
    - e.g. $\mathcal{P}$ is the set of all alpha carbon atoms in the molecule IgG1, each alpha carbon being a unit
- for simplicity, we often take $\mathcal{P} = \{1, \ldots, N\}$ where N is the (finite) cardinality of $\mathcal{P}$
- of course in some cases, it may be easier to identify the population and units **before** assigning labels to each unit

Problem: Populations and units

More generally, a **population** $\mathcal{P}$ is a set of identifiable **units** u:

- e.g. each alpha carbon in the molecule IgG1 is a **unit** in a population of size 1,556
- in practice, populations $\mathcal{P}$ are almost always **finite**
- units are unique and distinct from one another
- u can be thought of simply as a unique key and so can be represented by any identifiable and unique label for each unit in the population
  - e.g. the row number/label printed out by `head(igg1)`
- of course in some cases, it may be easier to identify the population and units **before** identifying individual units with unique labels
  - e.g. $\mathcal{P}$ is the set of all alpha carbon atoms in the molecule IgG1, each alpha carbon being a unit
- for simplicity, we often take $\mathcal{P} = \{1, \ldots, N\}$ where N is the (finite) cardinality of $\mathcal{P}$
- of course in some cases, it may be easier to identify the population and units **before** assigning labels to each unit

Notation: Populations will be distinguished from one another by using subscripts as in $\mathcal{P}_{IgG1}$.

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

More generally, we imagine **variates** to be functions $x(u)$, $y(u)$, and $z(u)$ which when called on any unit u return its value for that coordinate.

UNIVERSITY OF WATERLOO

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

More generally, we imagine **variates** to be functions $x(u)$, $y(u)$, and $z(u)$ which when called on any unit u return its value for that coordinate. That is, variables in `igg1` simply record values obtained by evaluating the corresponding variate on each unit u in $\mathcal{P}$.

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

More generally, we imagine **variates** to be functions $x(u)$, $y(u)$, and $z(u)$ which when called on any unit u return its value for that coordinate. That is, variables in `igg1` simply record values obtained by evaluating the corresponding variate on each unit u in $\mathcal{P}$. For example,

- `igg1$x` records values of $x(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$,
- `igg1$y` records values of $y(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$, and
- `igg1$z` records values of $z(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$.

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

More generally, we imagine **variates** to be functions $x(u)$, $y(u)$, and $z(u)$ which when called on any unit u return its value for that coordinate. That is, variables in `igg1` simply record values obtained by evaluating the corresponding variate on each unit u in $\mathcal{P}$. For example,

- `igg1$x` records values of $x(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$,
- `igg1$y` records values of $y(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$, and
- `igg1$z` records values of $z(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$.

The same is true for the remaining variables in `igg1`: recordType, name, residue, chainID, residueSequenceNum, residueName, group.

Problem: Units and variates

The data frame `igg1` also has 10 columns, each being a variable recording its value for every individual alpha carbon (unit) in the data frame.

For example, the three dimensional geometric location of the ith alpha carbon is recorded as the ith value of the variables `x`, `y`, and `z`.

More generally, we imagine **variates** to be functions $x(u)$, $y(u)$, and $z(u)$ which when called on any unit u return its value for that coordinate. That is, variables in `igg1` simply record values obtained by evaluating the corresponding variate on each unit u in $\mathcal{P}$. For example,

- `igg1$x` records values of $x(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$,
- `igg1$y` records values of $y(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$, and
- `igg1$z` records values of $z(u)$ for $u \in \{u_1, u_2, \ldots, u_{1556}\}$.

The same is true for the remaining variables in `igg1`: recordType, name, residue, chainID, residueSequenceNum, residueName, group. Each records the values of these **variates** for the units in our data set, namely $u \in \{u_1, u_2, \ldots, u_{1556}\}$.

Problem: On variates

A **variate** is just

- some *function* on any unit u

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259
  - x_u could take any **real** value, but is likely restricted to be in some finite real interval $[a, b]$ about 0

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259
  - x_u could take any **real** value, but is likely restricted to be in some finite real interval $[a, b]$ about 0
  - it follows that there are an **uncountably infinite** number of possible horizontal locations x between a and b.

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259
  - x_u could take any **real** value, but is likely restricted to be in some finite real interval $[a, b]$ about 0
  - it follows that there are an **uncountably infinite** number of possible horizontal locations x between a and b.
  - in such cases, we call $x = x()$ a **continuous** variate.

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259
  - x_u could take any **real** value, but is likely restricted to be in some finite real interval $[a, b]$ about 0
  - it follows that there are an **uncountably infinite** number of possible horizontal locations x between a and b.
  - in such cases, we call $x = x()$ a **continuous** variate.
  - this is a **ratio** scale variate since the ratio of any two values is meaningful

Problem: On variates

A **variate** is just

- some *function* on any unit u
- with *domain* $\mathcal{P}$ and
- the set of all possible values which that variate can take as its *range*

For example, for each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the x coordinate of its 3D location is $x(u)$, or simply x_u where $x_1 =$ `igg1$x[1]` = -62.259
    - x_u could take any **real** value, but is likely restricted to be in some finite real interval $[a, b]$ about 0
    - it follows that there are an **uncountably infinite** number of possible horizontal locations x between a and b.
    - in such cases, we call $x = x()$ a **continuous** variate.
    - this is a **ratio** scale variate since the ratio of any two values is meaningful
- similarly, the other two coordinates of the 3D locations $y(u)$ and $z(u)$ (or simply y_u and z_u) are also *continuous* and *ratio scale* variates.

UNIVERSITY OF WATERLOO

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)

UNIVERSITY OF WATERLOO

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and
  - are **categorical** variates in that not even the *order* of the values is meaningul (the values being only strings themselves)

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and
  - are **categorical** variates in that not even the *order* of the values is meaningul (the values being only strings themselves)
  - implemented in R as `factor` vectors, each having a finite set of `levels`

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and
  - are **categorical** variates in that not even the *order* of the values is meaningul (the values being only strings themselves)
  - implemented in R as `factor` vectors, each having a finite set of `levels`

Discrete variates where **only** the *order* of the possible values is meaningful are called **ordinal** variates

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and
  - are **categorical** variates in that not even the *order* of the values is meaningul (the values being only strings themselves)
  - implemented in R as `factor` vectors, each having a finite set of `levels`

Discrete variates where **only** the *order* of the possible values is meaningful are called **ordinal** variates

- e.g. a variate such as $preference(u) \in \{"hate", "dislike", "neutral", "like", "love"\}$

UNIVERSITY OF WATERLOO

Problem: More on variates

For each alpha carbon $u \in \mathcal{P}_{IgG1}$

- the `residueSequenceNum`
  - cannot take any real value between any two values in its range and so is called a **discrete** variate
  - can only take on finitely many variates and is therefore a **finite discrete** variate (there are also **infinite discrete** variates, e.g. counts)
  - is also an **interval** scaled variate since in addition to order, the difference (or interval) *between* values (in a chain) is meaningful (ratios are not)
  - is implemented in R as an `integer` vector
- the remaining variates, (e.g. $recordType(u)$, $chainID(u)$, etc.) are all
  - *finite discrete* variates having only a finite set of possible values and
  - are **categorical** variates in that not even the *order* of the values is meaningul (the values being only strings themselves)
  - implemented in R as `factor` vectors, each having a finite set of `levels`

Discrete variates where **only** the *order* of the possible values is meaningful are called **ordinal** variates

- e.g. a variate such as $preference(u) \in \{"hate", "dislike", "neutral", "like", "love"\}$
- there are no strictly ordinal variates in the `igg1` data (though several, *residueSequenceNum*, x, y, and z can each be ordered)

UNIVERSITY OF WATERLOO

Data: Realizations, observations, and variates

The first three rows of `igg1` are

```r
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                    group
## 1 Glutamic acid                   Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

Data: Realizations, observations, and variates

The first three rows of `igg1` are

```
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                    group
## 1 Glutamic acid                   Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** u

Data: Realizations, observations, and variates

The first three rows of igg1 are

```r
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                    group
## 1 Glutamic acid                   Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** u
- each column number (or unique variable name) identifies a **variate**

Data: Realizations, observations, and variates

The first three rows of igg1 are

```r
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                     group
## 1 Glutamic acid                    Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** *u*
- each column number (or unique variable name) identifies a **variate**
- the values in any column identify the **realizations** of the **variate** identified with that column *for all the units u*

Data: Realizations, observations, and variates

The first three rows of `igg1` are

```
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                    group
## 1 Glutamic acid                   Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** u
- each column number (or unique variable name) identifies a **variate**
- the values in any column identify the **realizations** of the **variate** identified with that column *for all the units* u
- the values in any row identify the **realizations** *of all variates* for that **unit** ;

Data: Realizations, observations, and variates

The first three rows of igg1 are

```r
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                    group
## 1 Glutamic acid                   Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** *u*
- each column number (or unique variable name) identifies a **variate**
- the values in any column identify the **realizations** of the **variate** identified with that column *for all the units u*
- the values in any row identify the **realizations** *of all variates* for that **unit** ;
- an entire row is often called an **observation** (typically multivariate) and an entire column (with some abuse of language) a **variate** (or even **variable**, given that's what it is called in R)

Data: Realizations, observations, and variates

The first three rows of igg1 are

```
head(igg1, n=3)
```

```
##   recordType name residue chainID residueSequenceNum       x      y       z
## 1       ATOM   CA     GLU       H                  1 -62.259 45.262 -16.149
## 2       ATOM   CA     VAL       H                  2 -60.766 48.666 -15.351
## 3       ATOM   CA     GLN       H                  3 -57.145 48.577 -16.631
##     residueName                     group
## 1 Glutamic acid                    Acidic
## 2        Valine Non-polar (hydrophobic)
## 3     Glutamine       Polar (uncharged)
```

This rectangular arrangement is a standard statistical representation where:

- each row number (or any other key unique to each row) represents a **unit** u
- each column number (or unique variable name) identifies a **variate**
- the values in any column identify the **realizations** of the **variate** identified with that column *for all the units u*
- the values in any row identify the **realizations** *of all variates* for that **unit** ;
- an entire row is often called an **observation** (typically multivariate) and an entire column (with some abuse of language) a **variate** (or even **variable**, given that's what it is called in R)

N.B. Some people refer to this standard arrangement and interpretation as a **tidy data** representation.

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

- the **size of the population** $N_{\mathcal{P}} = \#\mathcal{P}$, say, being the count of how many units are in that population and

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

- the **size of the population** $N_{\mathcal{P}} = \#\mathcal{P}$, say, being the count of how many units are in that population and
- the **set of labels** which identify the units, for example being $\{1, 2, \ldots, N_{\mathcal{P}}\}$ or perhaps a set of unique tags or memory locations for the units in $\mathcal{P}$

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

- the **size of the population** $N_{\mathcal{P}} = \#\mathcal{P}$, say, being the count of how many units are in that population and
- the **set of labels** which identify the units, for example being $\{1, 2, \ldots, N_{\mathcal{P}}\}$ or perhaps a set of unique tags or memory locations for the units in $\mathcal{P}$

A third variate which is also (nearly) always available is the **sequence of labels** which identify the units. Surprisingly, *the order in which the units appear in the data structure often proves to be meaningful*.

UNIVERSITY OF WATERLOO

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

- the **size of the population** $N_{\mathcal{P}} = \#\mathcal{P}$, say, being the count of how many units are in that population and
- the **set of labels** which identify the units, for example being $\{1, 2, \ldots, N_{\mathcal{P}}\}$ or perhaps a set of unique tags or memory locations for the units in $\mathcal{P}$

A third variate which is also (nearly) always available is the **sequence of labels** which identify the units. Surprisingly, *the order in which the units appear in the data structure often proves to be meaningful*.

Typically, there will be very many more of interest.

UNIVERSITY OF WATERLOO

Population attributes

Given any population, $\mathcal{P}$, it becomes of interest to find some meaningful and informative summaries of $\mathcal{P}$ based on its units and possibly on variates evaluated on units.

Any such summary will be called a **population attribute** and, as with variates, population attributes can be thought of as a function, this time of a population $\mathcal{P}$ rather than of a unit.
When we want to emphasise this we will write an attribute as $a(\mathcal{P})$.

There are always at least two possible summaries of any population:

- the **size of the population** $N_{\mathcal{P}} = \#\mathcal{P}$, say, being the count of how many units are in that population and
- the **set of labels** which identify the units, for example being $\{1, 2, \ldots, N_{\mathcal{P}}\}$ or perhaps a set of unique tags or memory locations for the units in $\mathcal{P}$

A third variate which is also (nearly) always available is the **sequence of labels** which identify the units. Surprisingly, *the order in which the units appear in the data structure often proves to be meaningful*.

Typically, there will be very many more of interest.

Population attributes – numerical

Population attributes can be **numerical** (possibly vector valued), **graphical** (i.e. any data visualization), or any combination of the two.

Population attributes – numerical

Population attributes can be **numerical** (possibly vector valued), **graphical** (i.e. any data visualization), or any combination of the two.

For example, a simple numerical attribute might be the percentage of alpha carbons that have `recordType == "HETATM"`

Population attributes – numerical

Population attributes can be **numerical** (possibly vector valued), **graphical** (i.e. any data visualization), or any combination of the two.

For example, a simple numerical attribute might be the percentage of alpha carbons that have `recordType == "HETATM"` or

```
prop <- with(igg1, sum(recordType == "HETATM") / length(recordType))
paste0(round(100 * prop), "%") # as a character string for printing
```

```
## [1] "14%"
```

Population attributes – numerical

Population attributes can be **numerical** (possibly vector valued), **graphical** (i.e. any data visualization), or any combination of the two.

For example, a simple numerical attribute might be the percentage of alpha carbons that have `recordType == "HETATM"` or

```r
prop <- with(igg1, sum(recordType == "HETATM") / length(recordType))
paste0(round(100 * prop), "%") # as a character string for printing
```

```
## [1] "14%"
```

Or, maybe, a two way table of counts for combinations of `chainID` and `group`

Population attributes – numerical

Population attributes can be **numerical** (possibly vector valued), **graphical** (i.e. any data visualization), or any combination of the two.

For example, a simple numerical attribute might be the percentage of alpha carbons that have `recordType == "HETATM"` or

```
prop <- with(igg1, sum(recordType == "HETATM") / length(recordType))
paste0(round(100 * prop), "%") # as a character string for printing
```

```
## [1] "14%"
```

Or, maybe, a two way table of counts for combinations of `chainID` and `group`

```
knitr::kable(with(igg1, table(chainID, group)))
```

	Acidic	Basic	Non-polar (hydrophobic)	Polar (uncharged)	Sugar
C	0	0	0	0	220
H	38	54	171	189	0
I	38	54	171	189	0
L	17	19	78	102	0
M	17	19	78	102	0

where some similarities and differences between chains are immediately apparent.

Chains `H` and `I` are "heavy", `L` and `M` "light", and `C` is a carbohydrate chain.

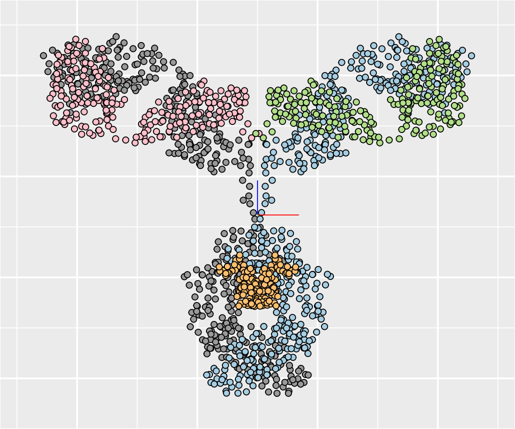
Population attributes – graphical

Alternatively, **graphical attributes** can sometimes provide complex summary information in a meaningful and comprehensible way.

Population attributes – graphical

Alternatively, **graphical attributes** can sometimes provide complex summary information in a meaningful and comprehensible way.

For example, as already seen, the geometric locations shown in an interactive 3D scatterplot can be very informative (here coloured by chain ID):



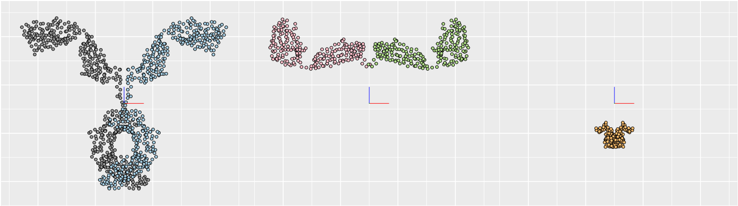
Population attributes – graphical

Interactive graphics, as in loon , make it very easy to construct informative graphical attributes by direct manipulation, as well as to save them for traditional publication:

```
heavyChain <- (igg1$chainID == "H") | (igg1$chainID == "I")
lightChain <- (igg1$chainID == "L") | (igg1$chainID == "M")
carbs <- (igg1$chainID == "C")
p3d["active"] <- heavyChain
p3d_heavy <- plot(p3d, draw = FALSE)
p3d["active"] <- lightChain
p3d_light <- plot(p3d, draw = FALSE)
p3d["active"] <- carbs
p3d_carbs <- plot(p3d, draw = FALSE)
# And plot these using grid graphics extra functionality
library(gridExtra)
# to arrange them in sequence
grid.arrange(p3d_heavy, p3d_light, p3d_carbs, nrow = 1)
```

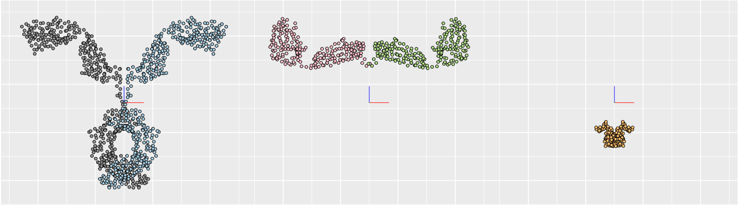
Population attributes – graphical

The three groups of chains, heavy, light, and carbohydrate:



Population attributes – graphical

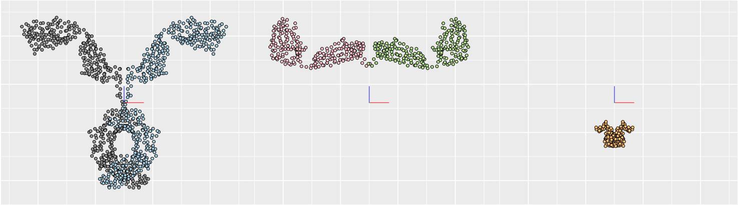
The three groups of chains, heavy, light, and carbohydrate:



Each of these three graphical attributes is an entire subset of the data.

Population attributes – graphical

The three groups of chains, heavy, light, and carbohydrate:

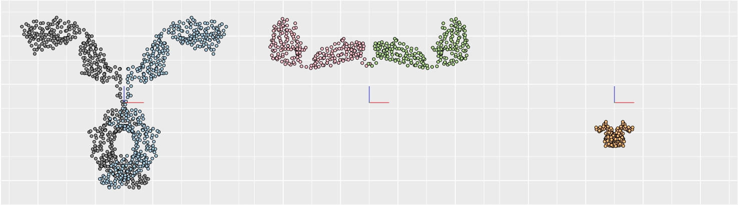


Each of these three graphical attributes is an entire subset of the data. Each is a presentation of four dimensional vectors:

$$< x(u), y(u), z(u), chainID(u) >$$

Population attributes – graphical

The three groups of chains, heavy, light, and carbohydrate:



Each of these three graphical attributes is an entire subset of the data. Each is a presentation of four dimensional vectors:

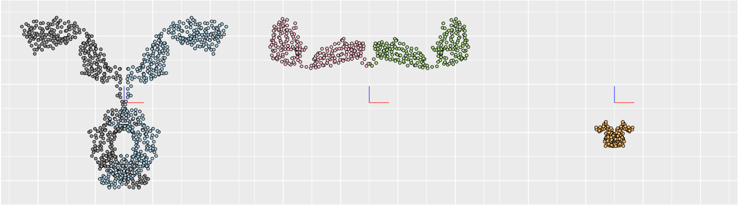
$$< x(u), y(u), z(u), chainID(u) >$$

for

1. $u \in \{u \ : \ u \in \mathcal{P} \ \text{ and } \ chainID(u) \in \{\text{"H"}, \text{"l"}\}\}$,

Population attributes – graphical

The three groups of chains, heavy, light, and carbohydrate:



Each of these three graphical attributes is an entire subset of the data. Each is a presentation of four dimensional vectors:

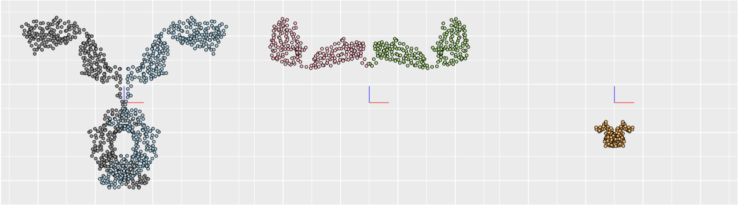
$$< x(u), y(u), z(u), chainID(u) >$$

for

1. $u \in \{u \;:\; u \in \mathcal{P}$ and $chainID(u) \in \{$"H", "I"$\}\}$,
2. $u \in \{u \;:\; u \in \mathcal{P}$ and $chainID(u) \in \{$"L", "M"$\}\}$,

Population attributes – graphical

The three groups of chains, heavy, light, and carbohydrate:



Each of these three graphical attributes is an entire subset of the data. Each is a presentation of four dimensional vectors:

$$< x(u), y(u), z(u), chainID(u) >$$

for

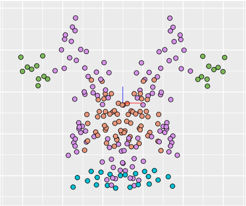
1. $u \in \{u \;:\; u \in \mathcal{P}$ and $chainID(u) \in \{$"H", "I"$\}\}$,
2. $u \in \{u \;:\; u \in \mathcal{P}$ and $chainID(u) \in \{$"L", "M"$\}\}$, and
3. $u \in \{u \;:\; u \in \mathcal{P}$ and $chainID(u) = "C"\}$.

Where $chainID(u)$ values are encoded by colour.

Population attributes – graphical

Or possibly zoom in on the carbohydrate chain coloured by `residue`:

```
p3d["active"] <- carbs
l_scaleto_active(p3d)
p3d["color"] <- igg1$residue
p3d["size"] <- 10
plot(p3d)
```



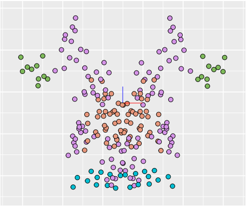
Which is now a presentation of five dimensional vectors:

$$< x(u), y(u), z(u), chainID(u), residue(u) >$$

Population attributes – graphical

Or possibly zoom in on the carbohydrate chain coloured by `residue`:

```
p3d["active"] <- carbs
l_scaleto_active(p3d)
p3d["color"] <- igg1$residue
p3d["size"] <- 10
plot(p3d)
```



Which is now a presentation of five dimensional vectors:

$$< x(u), y(u), z(u), chainID(u), residue(u) >$$

with

$$u \in \{u \;:\; u \in \mathcal{P} \;\text{ and }\; chainID(u) = "C"\}$$

and *residue*(u) values now encoded by colour.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

We design the study and collect the data to estimate, or test our preconceptions, about one or more attributes.

UNIVERSITY OF WATERLOO

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

We design the study and collect the data to estimate, or test our preconceptions, about one or more attributes. We are often trying to improve our understanding of these attributes by improved estimation and testing.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

We design the study and collect the data to estimate, or test our preconceptions, about one or more attributes. We are often trying to improve our understanding of these attributes by improved estimation and testing.

In **exploratory** investigations, the data are often already in hand. The purpose of the study is now to **discover** attributes by observing the structure found in the data.

UNIVERSITY OF WATERLOO

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

We design the study and collect the data to estimate, or test our preconceptions, about one or more attributes. We are often trying to improve our understanding of these attributes by improved estimation and testing.

In **exploratory** investigations, the data are often already in hand. The purpose of the study is now to **discover** attributes by observing the structure found in the data.

Having discovered interesting and meaningful attributes (especially those which were not anticipated), a follow up study would be designed to gather new data to **confirm** and **test** the attributes previously discovered.

Attributes: by design or by discovery

There may be several population attributes that one has in mind (and even defined) well **before** the data have even been collected, let alone examined.

This is typically the case whenever a study has been **designed** with the purpose to collect data so as to examine the attribute. The analysis then is sometimes called **confirmatory**.

We design the study and collect the data to estimate, or test our preconceptions, about one or more attributes. We are often trying to improve our understanding of these attributes by improved estimation and testing.

In **exploratory** investigations, the data are often already in hand. The purpose of the study is now to **discover** attributes by observing the structure found in the data.

Having discovered interesting and meaningful attributes (especially those which were not anticipated), a follow up study would be designed to gather new data to **confirm** and **test** the attributes previously discovered.

In either case, an attribute is a summary of $\mathcal{P}$ and as such it will always be of interest to examine how the well it does and does not describe all of the units it targets in its summary.

Quick numerical attributes

Some simple attributes are easily had (and are worth checking as a habit):

```r
summary(igg1)
```

```
##   recordType        name         residue     chainID residueSequenceNum
##  ATOM   :1336   CA      :1336   SER    :178   C:220   Min.   :  1.0
##  HETATM: 220    C1      :  18   VAL    :122   H:452   1st Qu.: 85.0
##                 C2      :  18   NAG    :112   I:452   Median :279.5
##                 C3      :  18   THR    :106   L:216   Mean   :301.2
##                 C4      :  18   PRO    :102   M:216   3rd Qu.:522.0
##                 C5      :  18   GLY    : 98           Max.   :716.0
##                 (Other): 130   (Other):838
##        x                   y                 z
##  Min.   :-71.18000   Min.   :-65.93   Min.   :-27.45500
##  1st Qu.:-17.32575   1st Qu.:-23.17   1st Qu.: -9.69500
##  Median : -0.01650   Median : 35.71   Median :  0.01050
##  Mean   : -0.00268   Mean   : 16.56   Mean   :  0.00856
##  3rd Qu.: 17.30550   3rd Qu.: 52.65   3rd Qu.:  9.68825
##  Max.   : 71.20500   Max.   : 75.38   Max.   : 27.52100
##
##             residueName                         group
##  Serine             :178   Acidic                  :110
##  Valine             :122   Basic                   :146
##  N-acetylglucosamine:112   Non-polar (hydrophobic):498
##  Threonine          :106   Polar (uncharged)       :582
##  Proline            :102   Sugar                   :220
##  Glycine            : 98
##  (Other)            :838
```

Each variate is given its own two columns of `name : value` pairs.

Quick numerical attributes

Some simple attributes are easily had (and are worth checking as a habit):

```r
summary(igg1)
```

```
##   recordType         name          residue      chainID residueSequenceNum
##  ATOM   :1336   CA      :1336   SER     :178   C:220   Min.   :  1.0     
##  HETATM: 220    C1      :  18   VAL     :122   H:452   1st Qu.: 85.0     
##                 C2      :  18   NAG     :112   I:452   Median :279.5     
##                 C3      :  18   THR     :106   L:216   Mean   :301.2     
##                 C4      :  18   PRO     :102   M:216   3rd Qu.:522.0     
##                 C5      :  18   GLY     : 98           Max.   :716.0     
##                 (Other): 130   (Other):838                              
##        x                  y                 z            
##  Min.   :-71.18000   Min.   :-65.93   Min.   :-27.45500  
##  1st Qu.:-17.32575   1st Qu.:-23.17   1st Qu.: -9.69500  
##  Median : -0.01650   Median : 35.71   Median :  0.01050  
##  Mean   : -0.00268   Mean   : 16.56   Mean   :  0.00856  
##  3rd Qu.: 17.30550   3rd Qu.: 52.65   3rd Qu.:  9.68825  
##  Max.   : 71.20500   Max.   : 75.38   Max.   : 27.52100  
##                                                          
##              residueName                        group    
##  Serine             :178   Acidic                  :110  
##  Valine             :122   Basic                   :146  
##  N-acetylglucosamine:112   Non-polar (hydrophobic):498  
##  Threonine          :106   Polar (uncharged)       :582  
##  Proline            :102   Sugar                   :220  
##  Glycine            : 98                                 
##  (Other)            :838                                 
```

Each variate is given its own two columns of `name : value` pairs.

- Categorical variates show counts of values.

Quick numerical attributes

Some simple attributes are easily had (and are worth checking as a habit):

```r
summary(igg1)
```

```
##   recordType        name         residue     chainID residueSequenceNum
##  ATOM  :1336   CA      :1336   SER    :178   C:220   Min.   :  1.0     
##  HETATM: 220   C1      :  18   VAL    :122   H:452   1st Qu.: 85.0     
##                C2      :  18   NAG    :112   I:452   Median :279.5     
##                C3      :  18   THR    :106   L:216   Mean   :301.2     
##                C4      :  18   PRO    :102   M:216   3rd Qu.:522.0     
##                C5      :  18   GLY    : 98           Max.   :716.0     
##                (Other): 130   (Other):838                             
##        x                   y                 z            
##  Min.   :-71.18000   Min.   :-65.93   Min.   :-27.45500  
##  1st Qu.:-17.32575   1st Qu.:-23.17   1st Qu.: -9.69500  
##  Median : -0.01650   Median : 35.71   Median :  0.01050  
##  Mean   : -0.00268   Mean   : 16.56   Mean   :  0.00856  
##  3rd Qu.: 17.30550   3rd Qu.: 52.65   3rd Qu.:  9.68825  
##  Max.   : 71.20500   Max.   : 75.38   Max.   : 27.52100  
##                                                          
##               residueName                      group    
##  Serine             :178   Acidic                 :110  
##  Valine             :122   Basic                  :146  
##  N-acetylglucosamine:112   Non-polar (hydrophobic):498  
##  Threonine          :106   Polar (uncharged)      :582  
##  Proline            :102   Sugar                  :220  
##  Glycine            : 98                                
##  (Other)            :838                                
```

Each variate is given its own two columns of `name : value` pairs.

- Categorical variates show counts of values.
- Numeric variates show traditional summary statistics of that variate's values.

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ... the quartiles $Q(1/4)$ and $Q(3/4)$?

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ...the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale:

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ... the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale: range = max - min

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ...the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale: range = max - min, IQR = interquartile range = $Q(3/4) - Q(1/4)$

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ... the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale: range = max - min, IQR = interquartile range = $Q(3/4) - Q(1/4)$
- Measures of symmetry:

UNIVERSITY OF WATERLOO

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ... the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale: range = max - min, IQR = interquartile range = $Q(3/4) - Q(1/4)$
- Measures of symmetry: ratio of $[Q(3/4) - Q(1/2)]$ to $[Q(1/2) - Q(1/4)]$, ...

Where's Waldo? Numerical attributes

What can we learn about the distribution of the values of these variates from these numbers?

- Measures of location: mean, median or $Q(0.5)$, ... the quartiles $Q(1/4)$ and $Q(3/4)$?
- Measures of spread/variation/scale: range = max - min, IQR = interquartile range = $Q(3/4) - Q(1/4)$
- Measures of symmetry: ratio of $[Q(3/4) - Q(1/2)]$ to $[Q(1/2) - Q(1/4)]$, ...

Exercise: consider what happens to each of these measures when any variate y is transformed to $z = ay + b$ for two non-zero constants a and b.

Quick graphical attributes

Similarly, in R , simple graphical attributes are also easily had (and worth checking as a habit).

Quick graphical attributes

Similarly, in R , simple graphical attributes are also easily had (and worth checking as a habit).

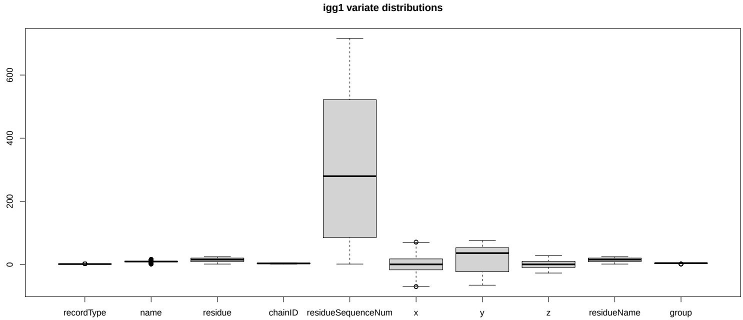
First, `boxplot()` will give graphical attributes of the **distribution** of each variate on *the same scale*

Quick graphical attributes

Similarly, in R , simple graphical attributes are also easily had (and worth checking as a habit).

First, boxplot() will give graphical attributes of the **distribution** of each variate on *the same scale*

```
boxplot(igg1, main = "igg1 variate distributions", col = "lightgrey")
```

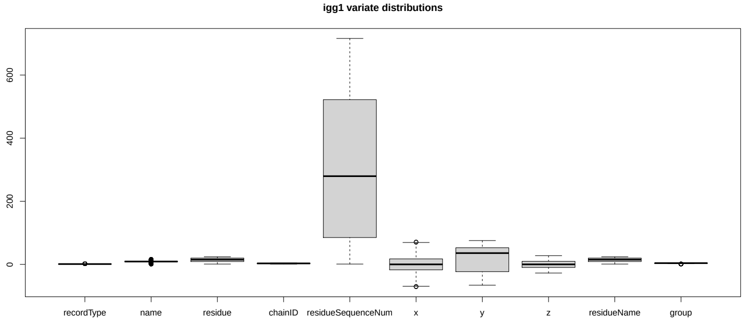


Quick graphical attributes

Similarly, in R , simple graphical attributes are also easily had (and worth checking as a habit).

First, boxplot() will give graphical attributes of the **distribution** of each variate on *the same scale*

```
boxplot(igg1, main = "igg1 variate distributions", col = "lightgrey")
```



Which is not that informative for most of the variates since they are categorical and boxplots are designed for continuous variates. Nevertheless, like summary() it gives a quick sense of the variates and the extent of their values.

There are other displays better suited to categorical variates.

Graphical attributes for categorical variates

Similarly, we might look at graphical attributes to summarize the distribution of values for each categorical variate.

Graphical attributes for categorical variates

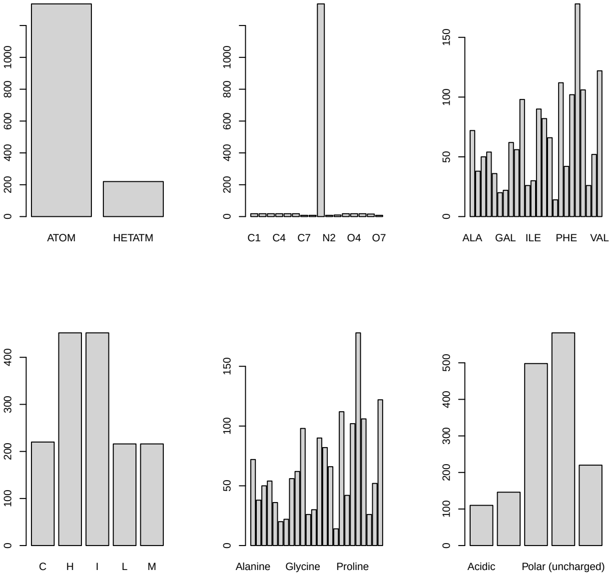
Similarly, we might look at graphical attributes to summarize the distribution of values for each categorical variate.

A bar plot for each:

```
isCatVar <- sapply(names(igg1), FUN  = function(name) is.factor(igg1[,name]))
catVars <- names(igg1)[isCatVar]
nrows <- floor(sqrt(length(catVars)))
ncols <- ceiling(sqrt(length(catVars)))
savePar <- par(mfrow = c(nrows, ncols))
for (var in catVars) {
    counts <- summary(igg1[,var])
    vals <- levels(igg1[,var])
    barplot(counts, names.arg = vals, col="lightgrey")
}
par(savePar)
```

Graphical attributes for categorical variates

A bar plot for each:



UNIVERSITY OF WATERLOO

Interactive graphical attributes for categorical variates

For exploratory work, it would be better if these were interactive.

```
isCatVar <- sapply(names(igg1), FUN  = function(name) is.factor(igg1[,name]))
catVars <- names(igg1)[isCatVar]
# Could simply have each plot in a separate window
# or in a single window as shown here
nrows <- floor(sqrt(length(catVars)))
ncols <- ceiling(sqrt(length(catVars)))

barplotWindow <- tktoplevel()   # THE WINDOW
row <- 0
col <- 0
for (var in catVars) {
    barplot <- l_hist(igg1[,var],
                      linkingGroup = "igg1",
                      title = var,
                      parent = barplotWindow)
    if (col >= ncols){
        row <- row + 1
        col <- 0}
    tkgrid(barplot, row = row, column = col, sticky = "nesw")
    col <- col + 1}
# Configure columns to resize with window
for (col in 0:(ncols-1)){tkgrid.columnconfigure(barplotWindow, col, weight = 1)}
# Configure rows to resize with window
for (row in 0:(nrows-1)){tkgrid.rowconfigure(barplotWindow, row, weight = 1)}
# Add a title
tktitle(barplotWindow) <- "Counts for factors"
```

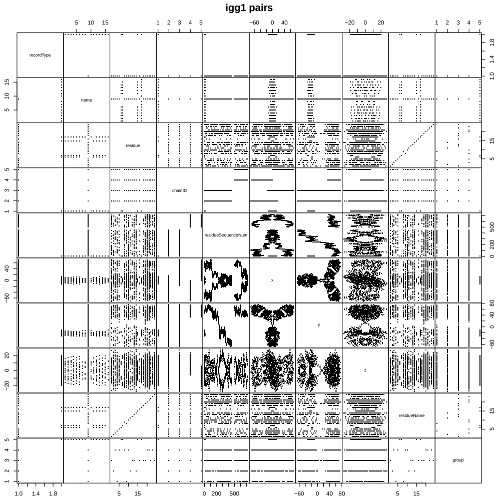
Quick graphical attributes - two dimensional

In R , there are also simple graphical attributes easily had for pairs of variates (and worth checking as a habit, provided there aren't too many).

Quick graphical attributes - two dimensional

In R , there are also simple graphical attributes easily had for pairs of variates (and worth checking as a habit, provided there aren't too many).

```
plot(igg1, gap = 0, pch = ".", col = "black", main = "igg1 pairs")
```



Might be better to restrict consideration just to those that variates that are not factors.

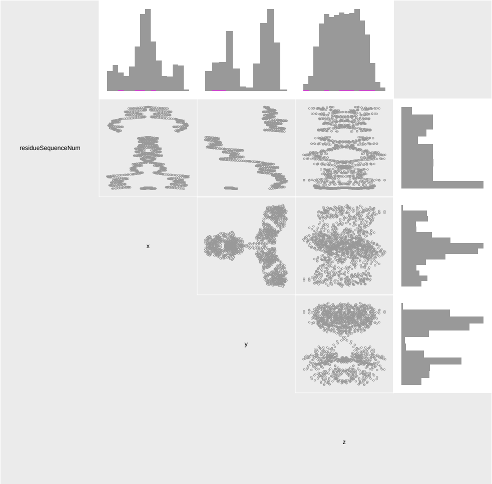
Quick graphical attributes - two dimensional interactive

An interactive version is available in loon via `l_pairs()`

Quick graphical attributes - two dimensional interactive

An interactive version is available in loon via `l_pairs()`

```
isCtsVar <- sapply(names(igg1), FUN  = function(name) !is.factor(igg1[,name]))
ctsVars <- names(igg1)[isCtsVar]
pp <- l_pairs(igg1[,ctsVars], glyph = "ocircle", size = 1,
              showHistograms = TRUE,
              linkingGroup = "igg1",  title = "Continuous pairs")
plot(pp)
```



Problem - Visible minorities in Canada 2006

Recall the `minority` data from loon.data.

Questions:

- What are the units u?

Problem - Visible minorities in Canada 2006

Recall the `minority` data from loon.data.

Questions:

- What are the units u?
- What are the variates u

Problem - Visible minorities in Canada 2006

Recall the `minority` data from loon.data.

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?

UNIVERSITY OF WATERLOO

Problem - Visible minorities in Canada 2006

Recall the `minority` data from loon.data.

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?
- What population attribute(s) are of interest?

Problem - Visible minorities in Canada 2006

Recall the `minority` data from loon.data.

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?
- What population attribute(s) are of interest?
- Are there any other populations that might be of interest?

Problem - Motor Trend cars 1974

Recall the `mtcars` data from R .

Questions:

- What are the units u?

Problem - Motor Trend cars 1974

Recall the `mtcars` data from R .

Questions:

- What are the units u?
- What are the variates u

Problem - Motor Trend cars 1974

Recall the `mtcars` data from R .

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?

Problem - Motor Trend cars 1974

Recall the `mtcars` data from R .

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?
- What population attribute(s) are of interest?

Problem - Motor Trend cars 1974

Recall the `mtcars` data from R .

Questions:

- What are the units u?
- What are the variates u
- What is the population $\mathcal{P}$?
- What population attribute(s) are of interest?
- Are there any other populations that might be of interest?